http://www.codingdefined.com/search/label/TypeScript?&max-results=7

**TYPESCRIPT**

## 1. What is TypeScript?

By definition, “TypeScript is JavaScript for application-scale development.”

TypeScript is a strongly typed, object oriented, compiled language. It was designed by **Anders Hejlsberg** (designer of C#) at Microsoft. TypeScript is both a language and a set of tools. TypeScript is a typed superset of JavaScript compiled to JavaScript. In other words, TypeScript is JavaScript plus some additional features.

2. Type the following command in the terminal window to install TypeScript.

> npm install -g typescript

3. A TypeScript program is composed of −

* Modules
* Functions
* Variables
* Statements and Expressions
* Comments

var message:string = "Hello World"

console.log(message)

## 4. TypeScript and Object Orientation

TypeScript is Object-Oriented JavaScript. Object Orientation is a software development paradigm that follows real-world modelling. Object Orientation considers a program as a collection of objects that communicate with each other via mechanism called methods. TypeScript supports these object oriented components too.

* **Object** − An object is a real time representation of any entity. According to Grady Brooch, every object must have three features −
  + **State** − described by the attributes of an object
  + **Behavior** − describes how the object will act
  + **Identity** − a unique value that distinguishes an object from a set of similar such objects.
* **Class** − A class in terms of OOP is a blueprint for creating objects. A class encapsulates data for the object.
* **Method** − Methods facilitate communication between objects.

**Example: TypeScript and Object Orientation**

class Greeting {

greet():void {

console.log("Hello World!!!")

}

}

var obj = new Greeting();

obj.greet();

The above example defines a class *Greeting*. The class has a method *greet ()*. The method prints the string “Hello World” on the terminal. The **new** keyword creates an object of the class (obj). The object invokes the method *greet ()*.

On compiling, it will generate following JavaScript code.

//Generated by typescript 1.8.10

var Greeting = (function () {

function Greeting() {

}

Greeting.prototype.greet = function () {

console.log("Hello World!!!");

};

return Greeting;

}());

var obj = new Greeting();

obj.greet()

The output of the above program is given below −

Hello World!!!

**5.TypeScript Types:**

## 1.The Any type:

The **any** data type is the super type of all types in TypeScript. It denotes a dynamic type. Using the **any** type is equivalent to opting out of type checking for a variable.

## 2. Built-in types:

## Number, String, Boolaen, null, undefined

### 3. User-defined Types

User-defined types include Enumerations (enums), classes, interfaces, arrays, and tuple.

### 6. Variables in TypeScript

var name:string = "John";

var score1:number = 50;

var score2:number = 42.50

var sum = score1 + score2

console.log("name"+name)

console.log("first score: "+score1)

console.log("second score: "+score2)

console.log("sum of the scores: "+sum)

**Interface:** In general, an interface is a device or a system that unrelated entities use to interact. According to this definition, a remote control is an interface between you and a television set, the English language is an interface between two people, and the protocol of behavior enforced in the military is the interface between people of different ranks.

Within the Java programming language, an [*interface* (in the glossary)](javascript:var%20meth=openWin;%20meth('interface');) is a type, just as a class is a type. Like a class, an interface defines methods. Unlike a class, an interface never implements methods; instead, classes that implement the interface implement the methods defined by the interface. A class can implement multiple interfaces.